

AMENDMENT

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1. (Currently Amended) A method of converting a call flow for a spoken dialog service into a state-based representation, the method ~~causing a computing device to perform steps~~ comprising:

walking a call flow for a spoken dialog service and converting each page of the call flow into a rule of a higher level representation of the call flow the higher level representation having a syntax used to specify grammars of a programming language, wherein the higher level representation differs from an XML format,~~wherein a name of at least one state transition in the higher level representation includes a suffix associated with a special meaning;~~

augmenting, via a processor, the higher level representation with terminal symbols representing dynamically typed state variable assignments and comparisons associated with decision and computation shapes in the call flow to yield an augmented higher level representation, wherein a name of at least one state transition in the augmented higher level representation includes a suffix associated with a special meaning; and

converting, via the processor, the higher level representation into a state-based representation.

2. (Previously Presented) The method of claim 1, wherein the higher level representation is a context-free grammar representation.

3. (Currently Amended) The method of claim [[2]] 1, wherein the higher level representation ~~context free grammar notation~~ is a Backus-Naur Form (BNF).
4. (Currently Amended) The method of claim [[3]] 1, wherein the state-based representation is a finite state machine (FSM).
5. (Previously Presented) The method of claim 3, wherein the step of walking the call flow and converting each page to a BNF occurs automatically via a computing device.
6. (Original) The method of claim 4, wherein the step of augmenting the BNF with terminal symbols occurs automatically via a computing device.
7. (Original) The method of claim 1, wherein the call flow comprises at least one page having a set of shapes having specific meanings.
8. (Previously Presented) The method of claim 7, wherein the set of shapes having special meanings comprises at least: parallelograms representing rules, lines representing dialog inputs, rectangular boxes representing dialog outputs or actions, diamonds representing Boolean decision functions, hexagrams representing calculation and assignment functions and annotation shapes representation comments.
9. (Original) The method of claim 4, wherein a grammar compiler is used to convert the BNF into the FSM.

10. (Original) The method of claim 9, wherein the FSM may be used by at least one spoken dialog tool to perform generation and testing functions associated with a spoken dialog service.

11. (Original) The method of claim 1, wherein the rule comprises terminal symbols comprising the names used to label shapes and transitions of the call flow.

12. (Currently Amended) The method of claim 1, further ~~causing the computing device to perform steps~~ comprising generating a unique terminal symbol in the higher level representation that shadows each rule, input, output, decision and calculation within the call flow.

13. - 20. (Canceled)

21. (Currently Amended) A system for converting a call flow for a spoken dialog service into a state-based representation, the system comprising:

a processor;

a first module configured for controlling the processor to walk a call flow for a spoken dialog service and convert each page of the call flow into a rule of a higher level representation of the call flow the higher level representation having a syntax used to specify grammars of a programming language wherein the higher level representation differs from an EXM format;
~~wherein a name of at least one state transition in the higher level representation includes a suffix associated with a special meaning;~~

a second module configured for controlling the processor to augment the higher level representation with terminal symbols representing dynamically typed state variable assignments and comparisons associated with decision and computation shapes in the call flow to yield an

augmented higher level representation, wherein a name of at least one state transition in the augmented higher level representation includes a suffix associated with a special meaning; and
a third module configured for controlling the processor to convert the higher level representation into a state-based representation.

22. (Previously Presented) The system of claim 21, wherein the higher level representation is a context-free grammar representation.

23. (Currently Amended) The system of claim ~~[[22]]~~ 21, wherein the higher level representation ~~context free grammar notation~~ is a Backus-Naur Form (BNF).

24. (Currently Amended) The system of claim ~~[[23]]~~ 21, wherein the state-based representation is a finite state machine (FSM).

25. (Currently Amended) The system of claim 23, wherein the first module ~~configured to walk the call flow and convert each page to a BNF~~, further walks the call flow and converts each page to a BNF automatically.

26. (Currently Amended) The system of claim 24, wherein the second module ~~configured to augment the BNF with terminal symbols~~ further augments the BNF automatically.

27. (Previously Presented) The system of claim 21, wherein the call flow comprises at least one page having a set of shapes having specific meanings.

28. (Previously Presented) The system of claim 27, wherein the set of shapes having special meanings comprises at least: parallelograms representing rules, lines representing dialog inputs, rectangular boxes representing dialog outputs or actions, diamonds representing Boolean decision functions, hexagrams representing calculation and assignment functions and annotation shapes representation comments.

29. (Previously Presented) The system of claim 24, wherein a grammar compiler is used to convert the BNF into the FSM.

30. (Previously Presented) The system of claim 29, wherein the FSM may be used by at least one spoken dialog tool to perform generation and testing functions associated with a spoken dialog service.

31. (Previously Presented) The system of claim 21, wherein the rule comprises terminal symbols comprising the names used to label shapes and transitions of the call flow.

32. (Currently Amended) The system of claim 31, further comprising a fourth module configured to generate a unique terminal symbol in the higher level representation that shadows each rule, input, output, decision and calculation within the call flow.